**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**\*\*\*\*\*\*\*\***

BÁO CÁO HW06

**Học phần: ITSS Software Development**

**Giảng viên hướng dẫn:** Trịnh Tuấn Đạt

**Sinh viên thực hiện:** Dương Đức Huy

**Mã lớp:** 147730

**Nhóm số:** 16

**Hà Nội, tháng 5 năm 2024**

**1. Tách biệt logic và giao diện người dùng:**

* **Chỉnh sửa:**
  + Tạo một lớp riêng biệt cho logic quản lý đơn hàng (OrderManager) để xử lý tất cả các hoạt động liên quan đến đơn hàng như thêm, xóa, cập nhật.
  + Tách biệt phương thức xử lý sự kiện khi người dùng tạo đơn hàng vào lớp OrderManager.
* **Ý nghĩa:**
  + Giúp mã nguồn trở nên sạch sẽ hơn, dễ bảo trì và mở rộng.

**2. Sử dụng mẫu thiết kế Singleton:**

* **Chỉnh sửa:**
  + Áp dụng mẫu thiết kế Singleton cho lớp OrderManager để đảm bảo chỉ có một thể hiện của nó được tạo ra trong toàn bộ ứng dụng.
* **Ý nghĩa:**
  + Tránh tạo ra nhiều thể hiện không cần thiết của lớp OrderManager, giảm sự phụ thuộc và tăng tính mô-đun của ứng dụng.

**3. Tách biệt các lớp và sử dụng Dependency Injection:**

* **Chỉnh sửa:**
  + Tạo một lớp riêng biệt cho quản lý giao diện người dùng (SiteFrameController) để xử lý tất cả các hoạt động liên quan đến giao diện.
  + Sử dụng Dependency Injection để cung cấp thể hiện của OrderManager cho SiteFrameController.
* **Ý nghĩa:**
  + Tách biệt logic và giao diện giúp làm rõ ràng hóa mã nguồn và giảm sự phụ thuộc giữa các thành phần.

**4. Áp dụng Thiết kế theo nguyên tắc SOLID:**

* **Chỉnh sửa:**
  + Đảm bảo rằng lớp OrderManager tuân theo các nguyên tắc của SOLID như Single Responsibility Principle (mỗi lớp chỉ nên có một trách nhiệm).
  + Sử dụng Interface cho OrderManager để tuân thủ nguyên tắc Dependency Inversion Principle.
* **Ý nghĩa:**
  + Áp dụng các nguyên tắc SOLID giúp tăng tính mô-đun, dễ bảo trì và mở rộng của mã nguồn.

**5. Phân chia chức năng và sử dụng Design Patterns:**

* **Chỉnh sửa:**
  + Sử dụng Design Pattern như Factory Pattern để tạo đối tượng Order.
  + Phân chia chức năng thành các phương thức nhỏ hơn trong lớp OrderManager để tăng tính tái sử dụng và dễ bảo trì.
* **Ý nghĩa:**
  + Giảm sự phức tạp của mã nguồn, tăng khả năng mở rộng và bảo trì.